MACHINE RECOGNITION OF CURSIVE WRITING

L. D. EARNEST

The Mitre Corp., Bedford, Mass., USA

1. Introduction

Machine reading of handwritten script has been chosen as a problem because it is interesting, potentially useful, and seemingly more difficult than character recognition, yet (hopefully) more tractable than speech recognition. The design of an appropriate information processing system is our principal interest. The processes by which humans accomplish the feat are of interest only to the extent that they may reveal some "tricks of the trade" of our principal competition.

Some interesting work has been done on script recognition, given the motion of the writing instrument as a function of time [1]). The investigation reported below uses the somewhat smaller amount of information available from the written pattern itself.

Based on experience, the author feels best qualified to discuss how *not* to design a script recognition system, having seen many seemingly good ideas perish in the harsh experimental environment of his own handwriting. Following the conventions of reporting one's work, however, positive results will be emphasized.

While it is reasonable to approach the mechanical reading of printed matter on a character-by-character basis, this approach appears unfeasible for written matter. The unit of written information is the word, there being in general, no apparent way to segment a word into letters without first recognizing the word. Thus, schemes for recognizing isolated script characters appear to offer little help. In addition to combinatorial complexity, handwriting generally exhibits more sloppiness and variations of style than does handprinting. Clearly, we must use more powerful tools than the *template matching* schemes that have enjoyed some success in printed character recognition.

2. Conventions and Constraints

Known conventions and constraints on the handwriting process seem to offer some help, if we can utilize them effectively. For the class of writing under investigation, it is conventional to write more or less horizontally from left to right, with certain *small* letters (*a*, *c*, *e*, etc.) written within an envelope bounded by two imaginary (sometimes real) lines. Certain *high* and *low* letters conventionally penetrate upward and downward outside the envelope (e.g. *b*, *f*, and *g*). In order to take advantage of this convention, we must either constrain the writer to write between known lines, or must find a way to estimate these lines (or, perhaps, determine one line in each way). While our current experimental system requires that the writing be more or less horizontal, we have chosen not to constrain the writer otherwise, so the envelope bounds must be estimated by the system.

In a given language, only certain letter sequences may be permitted. A more stringent constraint is imposed if the vocabulary is assumed to be known and finite. Using the latter assumption, we approach script recognition as a problem of properly categorizing each script sample, then performing a succession of discriminative tests to yield a progressively shorter word list.

Although the syntactic, semantic, and higher order constraints on the language would seem to be applicable to our problem, we do not yet have any very good ideas on how to use them.

In the actual process of writing, the human hand and associated control system seem to find cycloid-like strokes easiest to produce. In fact, conventional script can be nicely synthesized by a properly chosen sequence of cycloid segments [2]). There also appear to be rules of formation for stroke sequences [3]). It is planned to use these properties.

Although it is planned to add certain adaptive processes to the system, it is by no means self-organizing (unless one includes the programmer as a system element, and even then it is debatable). The employment of the above constraints to achieve recognition is accomplished through three principal kinds of operation.

a) Input patterns are encoded as two-dimensional arrays of binary elements (i.e. Boolean matrices with ***0*** and ***1*** standing for *white* and *black*). Transformations and tests on these patterns are performed by a Boolean matrix processor.

b) Certain arithmetic operations and tests are performed, mostly in connection with modelling of the physical handwriting process.

c) Language contraints are imposed through symbol manipulation, which uses list processing techniques.

3. Experimental Facilities

The current experimental system has evolved over a two year period, first using the TX-0 computer at the Massachusetts Institute of Technology and more recently the M.I.T. Lincoln Laboratory TX-2. The latter machine has a number of unusual features, including a multi-sequenced, multi-configured central processor [4]) and a small magnetic thin film memory. The large dictionary being used is readily stored in the main core memories (which include one module of 2.4 million bits – the largest in the world). A variety of input/output devices facilitate on-line experimentation.

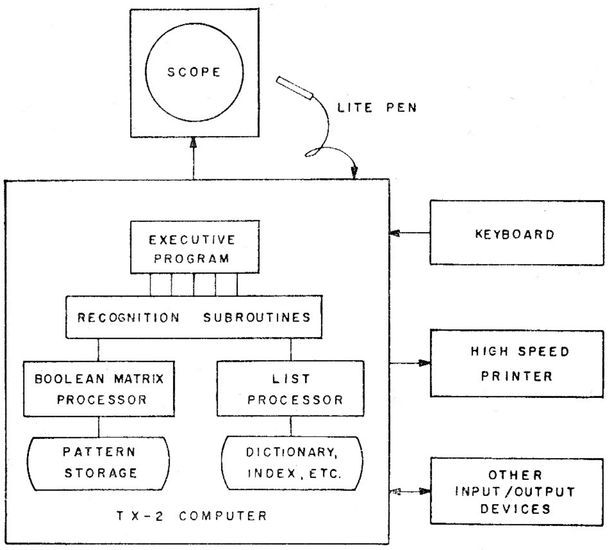

Fig. 1. Experimental system configuration

Even though the simulation of Boolean matrix operations is relatively time consuming, TX-2 executes the current word recognition program in about five seconds.

Although TX-2 has graphic and photographic input devices, it has been more convenient thus far to write script samples directly into the computer using a cathode-ray tube and *lite* pen (a photoelectric device whose binary output is sensed by the computer). A programmed display-feedback process is used to follow the motion of the pen over the surface of the cathode-ray tube. It is clear that in using this input technique we are deferring certain important difficulties in reading actual paperwork — such things as smudges, gross variations in line width, and the problem of separating closely spaced words. The *lite* pen does share at least one frailty with its real world counterpart — *ball point skip.*

The operator may select any of a number of operations (e.g., "clear the input matrix") by pointing the *lite* pen at appropriate control symbols that are displayed on the periphery of the scope. Results are generally displayed on the scope, with xerographic output as an alternative.

Internal manipulations of encoded images are performed by a Boolean matrix processor that is not unlike some that have been employed by other investigators on other pattern recognition problems [5]). An interpretive program simulates the matrix operations in response to single-address commands, an Operation Matrix playing a role analogous to the conventional accumulator. It is into this matrix that the *lite* pen writes, and its contents are normally displayed on the cathode-ray tube. The matrix size is variable — currently 100 × 176 elements.

Matrix commands include logical operations (AND and OR) optionally combined with inversion (NOT) and composite shifts in both dimensions. A connectivity command (expand) facilitates the extraction of individual line segments. Other instructions find the *X* and *Y* coordinates of pattern extremes, find the number of ***1***'s in any given row or column, permit the insertion of a ***1*** in any given element, and the insertion of ***0***'s between any given pair of *X* or *Y* coordinates.

Much of the work of comparing observed features of the pattern with the language constraints consists of symbol manipulation, and is accomplished by processing lists of words, characters, properties, and so forth. These functions are performed by a collection of subroutines written in machine language. Under hindsight, the use of a specialized list processing language and corresponding interpretive program would seem more appropriate.

The largest list is a 10 397 word dictionary that represents (approximately) those words that occur more often than three times per million words of typical English text [6]), omitting proper nouns. Another sizable list is an index to the dictionary based on a categorization of certain features in the script representation of each word.

4. Feature Extraction

For the existing recognition process, single word samples must be written more or less horizontally and no capital letters may be used. The only constraints on position and scale are those imposed by the available writing space (5″ × 9″) and the pattern quantization (0.05″ × 0.05″).

When recognition of a word is requested by the writer, a simple gap-filling program is first executed by the matrix processor, to compensate for *ball point skip.* Next, the number of ***1***'s in each row of the matrix is scanned from bottom to top to find the *Y*-coordinates of the first row having a ***1***-count above, then below, certain thresholds that are set by a measure of overall pattern density. These two values define the estimated envelope of the central letters (*a* and *e* in fig. 2).

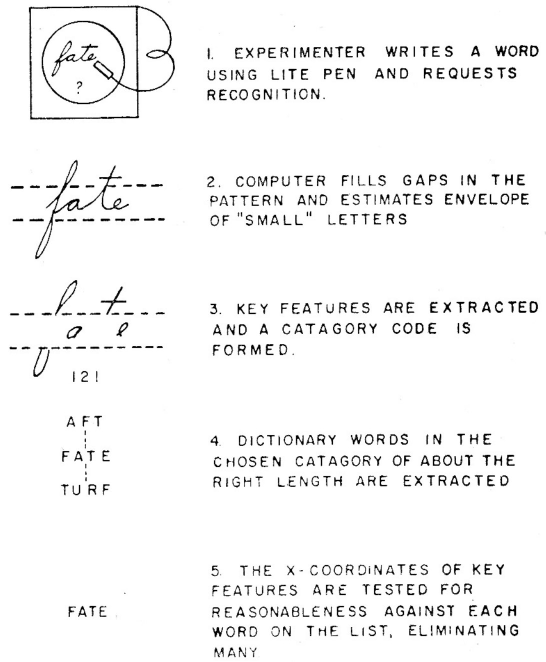

Fig. 2. Recognition sequence

While the present envelope detection scheme works well most of the time, it has difficultly with words having relatively few central letters (e.g. *lilt*). In an operational system it would be possible to detect and correct most of such difficulties by cross-checking the estimated envelopes of succeeding words of the same line.

Using the estimated envelope as a guide, a matrix program finds the following key features:

1. significant strokes above the envelope (as in *b*, *d*, *f*, etc.) and whether such strokes have a crossbar (i.e. *t*),
2. significant strokes below the envelope (as in *f*, *g*, *j*, etc.),
3. significant closed curves entirely within the envelope (always in *e*, sometimes in *a*, *b*, *d*, etc.);
4. the number of times the pattern crosses an axis midway between the envelope lines.

It should be mentioned that the feature detection programs are based on connectivity within a given region, so that when crossbars overlap adjacent tall letters (as in *little*), the overlapped sequence is detected as a single grand feature. This usually causes no subsequent difficulty, however.

The above features have been selected for their ease and reliability of detection, and currently constitute the only information that is available to the balance of the recognition process. Certain additional features are under consideration for inclusion.

5. Dictionary Partitioning

Once the key features have been identified, a 7-bit category code is constructed as follows: The highest order bit tells whether any crossbars were found, the next three bits tell the number of *high* strokes, and the last three tell the number of *low* strokes. Since the example of fig. 2 has a *t*, two high strokes, and one low, its octal code is 121. As noted above, tall letters adjacent to *t*'s may or may not be distinguished as separate features, depending on how the crossbars are drawn. To avoid difficulty, such words are filed under each possible catagory. For example, the word "little" is filed under 120, 130, and 140.

If the above categorization split the dictionary into equal sized blocks, there would be eighty-odd words in each block. Non-uniformity of the distribution causes the actual average (over the word population) to be about eight times this value.

The next step in the process is a comparison of the predicted number of axis crossings of each word in the chosen category with the observed value (within a certain tolerance). On the average, about half the words survive this test.

Finally, a crude model of the handwriting process steps across the pattern from left to right, checking the X-coordinates of extracted features against corresponding features (if any) in each word remaining on the list. While satisfactory words are required to have an appropriate letter corresponding to each central closed curve of the pattern, and vice versa for all *e*'s, the letters *a*, *b*, *d*, *g*, etc. are not required to be closed.

The above tests are all that have been programmed to date and are sometimes sufficient for unique recognition. In one series of tests using words selected at random from the dictionary, some 18 per cent were uniquely identified. The residual list contained 9 words or less in 65 per cent of the *successful* tests (more is said about the unsuccessful tests later). Unfortunately, those word catagories that contain few of the features being sensed are not well subdivided by the existing checks (catagory 000 being a particular offender). The resultant occasional long lists run the current average up to about 20 words. Even so, this represents a discrimination ratio of 500 to 1 over the dictionary as a whole, which does not seem bad for such simple tests.

Concerning the reliability of the process, five subjects who were unfamiliar with the system (and *vice-versa*) were recently asked to write words selected at random from the dictionary. Out of a total of 107 words, the computer correctly listed 65 (60 per cent success). Quite a number of failures appeared to be readily fixable, being consequences of style variations not considered in the preceding parameter optimization process. Some of the failures appeared to result from difficulties in manipulating the *lite* pen (which has a tendency to dribble "ink" when lifted).

On the other hand, the results of the above experiment were probably biased upward by an unplanned *teaching machine* effect. Some five seconds after the subject writes a word, he finds out whether or not the machine succeeded. Unfortunately, there is then a tendency for the writer to view the outcome as a measure of his own success and to modify his writing so as not to confuse the machine.

6. Future Forgery

It is believed that the present system can be improved substantially through refinements to the existing structure. At the same time, some more subtle discrimination technique would seem to be required if we wish to achieve generally unique recognition over a large vocabulary.

One possible approach is through *analysis-by-synthesis* [2,3]). Given a list of alternatives, such as one produced by the present system, the proposed process would synthesize each word stroke-by-stroke, adjusting the parameters of generation for best fit (each cycloidal stroke being characterized by some 4 or 5 parameters). The word(s) that match the pattern without having to use abnormal generation parameters would then be selected. It is interesting to note that this technique is potentially capable, not only of recognition, but of forgery as well. Given a handwriting sample, the distribution of parameters that characterize the writing style can be determined and can then be used to synthesize new words in the same style. (We shall not consider the potential of automatic personality analysis based on writing style.)

Leaping even further into the unknown, it does not seem unreasonable that mechanical reading devices may achieve lower error-rates than do human readers. While some people find it difficult to adapt to another person's handwriting, a well-fed machine could readily compare the strokes of a given sample with numerous similar sequences in previously recognized words from the same subject.

7. Conclusions

It appears that reliable mechanical recognition of cursive writing is not out of the question, though it has not yet been achieved. It will apparently be necessary to make the most of graphic, linguistic, and dynamic constraints on the writing process. Boolean matrix and list processing techniques offer convenient and powerful tools for executing the necessary processes.

8. Acknowledgements

Professors M. Eden and M. Halle of the Massachusetts Institute of Technology have contributed considerable inspiration and numerous specific ideas to the present work. The author is indebted to Professor J. B. Dennis, also of MIT, for his support of the early work and to W. A. Clark of the MIT Lincoln Laboratory for making available the experimental computing facilities.

9. References

1) Frishkopf, L. S. and L. D. Harmon: *Machine Reading of Cursive Script*. Information Theory (London 1960).

2) Eden, M.: *Handwriting and Pattern Recognition*. (to be published) IRE Trans. PGIT.

3) Eden, M. and M. Halle: *The Characterization of Cursive Writing*. Information Theory (London 1960).

4) Clark, W. A.: *The Lincoln TX-2 Computer Development* Proc. W. Joint Comp. Conf. (1957).

5) Unger, S. H.: *A Computer Oriented Toward Special Problems*. Proc. IRE **46** (Oct. 1958).

6) Thorndike and Lorge: "The Teacher's Word Book of 30 000 Words" (New York 1944).

ABSTRACTS

Some 10 000 of the more common English words have been categorized on the basis of salient features in their handwritten shapes. Script samples of words to be recognized are processed to determine their position and scale in both dimensions, and are categorized on the same basis as the dictionary. A list of properties is extracted from each word pattern and is compared with the corresponding dictionary entries to reduce ambiguity, sometimes yielding unique recognition, but more often a short list of words. For non-unique cases, further isolation is undertaken by attempting to synthesize and match each of the alternatives with the pattern under observation. The technique of finding the position and scale of the pattern, and its application in identifying and comparing the coordinates of key features with the dictionary entries is discussed.

Около 10 000 наиболее употребительных английских слов классифицируются на основе примечательных свойств их рукописных форм. Рукописные образцы слов, подлежащих распознаванию, обрабатываются для определения их положения и шкалы в обоих измерениях и каталогизируются на такой же основе, что и словари. Из каждой конфигурации слова извлекается перечень свойств и сравнивается с соответствующей строчкой словаря для уменьшения неопределенности. В результате получается иногда полное опознавание, но чаще составляется краткий список слов. Для случаев неопознания должна быть предпринята дальнейшая изоляция, чтобы синтезировать и "согласовать" каждую из альтернатив с наблюдаемой конфигурацией. Обсуждается метод нахождения положения и шкалы конфигурации и его применение в идентификации и сравнении координат ключевых особенностей со статьями словаря.

On a catalogué 10 000 des mots anglais les plus communs en se basant sur les caractères saillants de leurs formes manuscrites. Les exemples écrits des mots à reconnaître sont traités pour déterminer leur position et leur rang dans les deux dimensions, et ils sont catalogués de la même façon que le dictionnaire. Une liste de propriétés est extraite de chaque exemple de mot et elle est comparée aux entrées correspondantes du dictionnaire pour réduire l'ambiguïté; quelquefois, il se présente une reconnaissance unique, mais le plus souvent il y a une courte liste de mots. Pour les cas n'ayant pas une reconnaissance unique, on entreprend l'isolation ultérieure en essayant de synthétiser et d'adapter chaque alternative à l'eemple qui découle de l'observation. On discute la technique de recherche de la position et du rang de l'exemple, et ses applications pour l'identification et la comparaison des coordonnées de caractères-clé avec les entrées du dictionnaire.

Einige 10 000 der gebräuchlicheren englischen Wörter wurden auf Grund von hervorstehenden Merkmalen ihrer handgeschriebenen Formen in Kategorien geordnet. Schriftproben von zu erkennenden Wörtern werden einem Verfahren unterworfen, um Lage und Grössenverhältnisse in beiden Dimensionen zu bestimmen, und wie in einem Wörterbuch in Kategorien eingeteilt. Aus jedem Wort wird eine Liste von Eigenschaften extrahiert, die mit den entsprechenden Wörterbucheingängen verglichen wird, um Mehrdeutigkeit zu vermeiden, und die manchmal eindeutige Erkennung, aber viel öfter eine kurze Liste von Wörtern ergibt. In nicht eindeutigen Fällen, muss noch weiter getrennt werden, indem man mittels Synthese versucht, jede der Möglichkeiten mit dem beobachteten Muster zur Deckung zu bringen. Die Technik des Auffindens der Lage und der Grössenverhältnisse des Musters und ihre Anwendung bei der Identifikation und beim Vergleichen der Koordinaten der Schlüsselmerkmale mit den Wörterbucheingängen wird diskutiert.

Se han clasificado unas 10 000 palabras inglesas entre las más frecuentes, apoyandose en la base de los rasgos más destacados en sus formas escritas a mano. Las muestras de escrituras a ser reconocidas son tratadas para determinar su posición y escala en ambas dimensiones y se clasifican sobre la misma base que el diccionario. Se extrae de cada palabra una lista de propiedades y se compara con las correspondientes entradas en el diccionario para reducir la

ambigüedad. Algunas veces esta lista conduce a un único reconocimiento, pero es mas frecuente que corresponde a una lista de palabras. Para los casos no únicos, es preciso acometer un aislamiento más completo y se intenta sintetizar y emparejar cada una de las alternativas con el modelo en observación. Se discute la técnica para encontrar la posición y escala del modelo y su aplicación a la identificación y comparación de las coordenadas de los rasgos clave con las entradas del diccionario.

DISCUSSION

L. UHR *(USA)*. I tested the learning program described elsewhere on handwriting, asking a special subroutine to make very primitive tentative segmentations and recognize letter-by-letter. We got only a little better than 50% success. If we were to take advantage of the great redundancy in English, by using word or diagram frequencies we would appreciably improve these results. Incidentally, we found speech to be an easier problem than handwriting for our program. Why do you feel speech recognition is so much more difficult?

L. EARNEST *(USA)*. I have relied on the opinions of others that the speech segmentation problem is most difficult. It is interesting that you have found otherwise.

M. NADLER *(France)*. After all, if you get high recognition of strings of only 6 characters by your dictionary consultation technique, with Uhr's 50% recognition of individual letters, he should do even better on words.

L. EARNEST *(USA)*. It would be interesting if that could be done. A decision procedure would have to be devised to map from Uhr's letters to the dictionary. It is my experience that this kind of transformation is much easier to conceive than to realize.